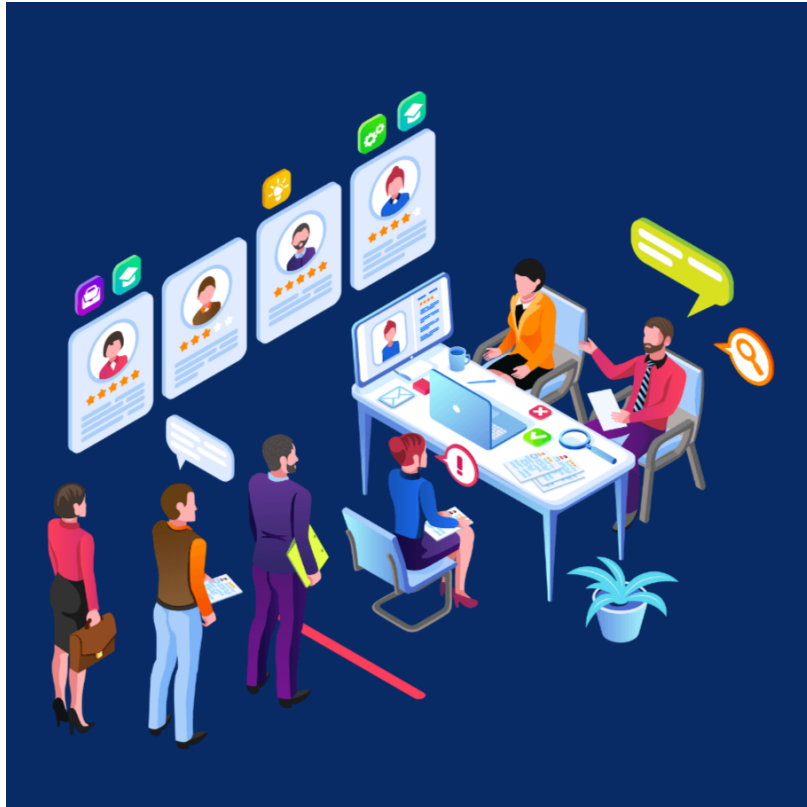


---

# Placement Cell Database

**Darshan Pathak**



---

## CASE STUDY

A College wants to develop a placement management system. Every year interviews are conducted in the college and students are placed into the companies. The college wants to create a database of the students, companies, interviews, and placed students. The selection criteria are based on the skills, i.e., if the skills acquired by the students and the skills required by the company match when the interview is scheduled, then the students would be placed.

## INTRODUCTION

The case study presented here outlines the development of a Placement Management for a college, focusing on the creation of a comprehensive database to facilitate the placement of students into companies based on their skills. This system aims to streamline the placement process and enhance data management. The case study involves the identification of main entities and relationships, the creation of an Entity-Relationship (ER) diagram, the design of a database schema, normalization, MySQL database creation, and the provision of SQL queries to demonstrate the functionality of the database. This practical application of database design and management serves as a valuable illustration of how such systems can be effectively implemented in an educational context.

### Activities to be performed

1. Identify the main entity types.
2. Identify the main relationship types and specify the multiplicity for each relationship.
3. Using answers (1) and (2), draw a single ER diagram to represent the data requirements.
4. Design a database schema for the above case.
5. Normalize the database up to 3NF.
6. Create a database [with necessary constraints] in MYSQL of schema model.
7. Write at least 25 queries based on the above model.

### Step-1: Identifying the main entity types for the Training and Placement Management System

In the above case study, we have observed the following entities about which the data is to be collected:

1. Students
2. Skills
3. Companies
4. Placements

---

## Step-2: Identifying the main relationship type for each entity

### 1. (Students - Skills)

One student has many skills, and many students have the same skill, so there will be a many-to-many relationship between them stored in StudentSkills table.



### 2. (Companies - Skills)

A company can require more than one skill, and a skill can be required by more than one company, so there will be a many-to-many relationship between them stored in CompanyReqSkills table.



### 3. (Students - Interviews)

A student can appear for interviews in many companies, and a company can interview more than one student, but the date of the interview and the interviewID will be different and a student can interview for a same company after adding some skills that are required by the company .

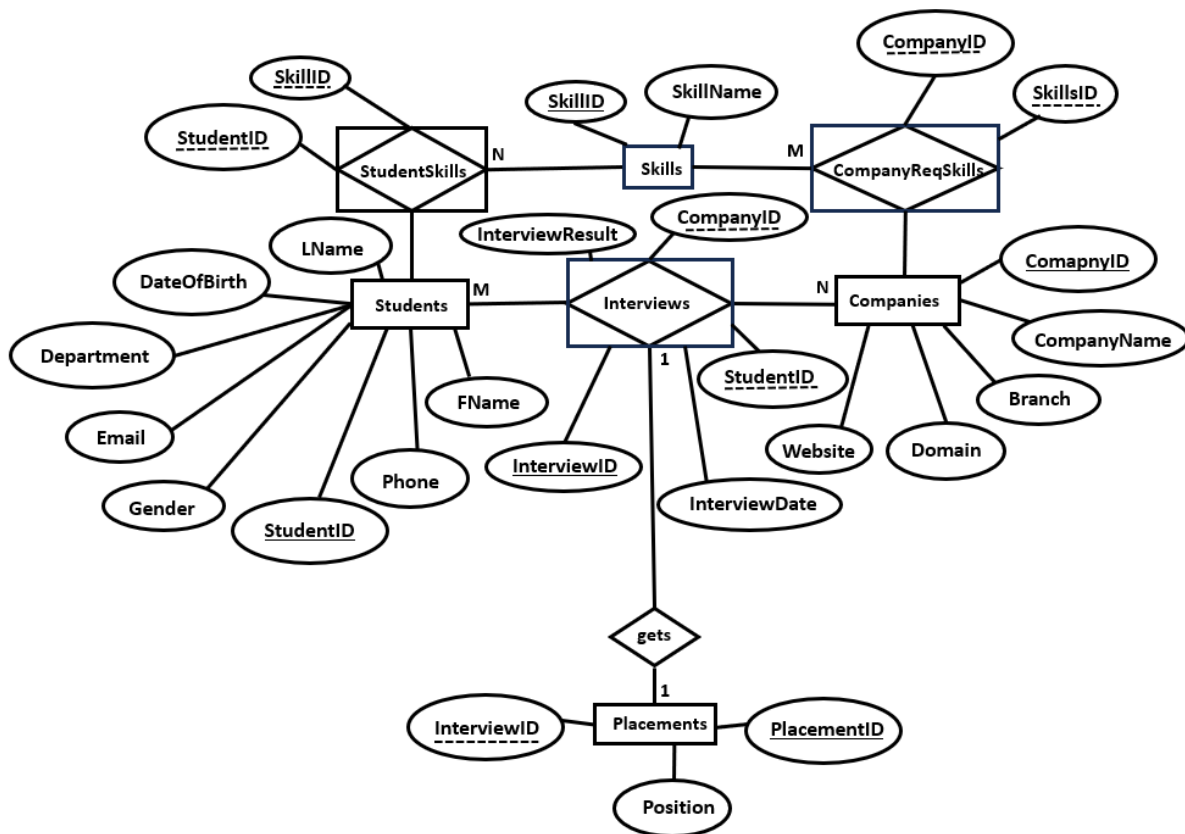


#### 4. (Interviews - Placements)

After appearing for an interview, one student can get placed only once, so there is a one-to-one relationship between interviews and placements.



Step-3: Combining all the steps, the ER diagram for the above case study is as follows



ER Diagram

---

## Step-4: Create database Schema

Creating a database schema of the above ER-Diagram.

- **STUDENT** (StudentID, FName, LName, DateOfBirth, Department, Email, Gender, Phone)
- **Companies** (CompanyID, CompanyName, Domain, Branch, Website)
- **Skills** (SkillID, SkillName)
- **Placements** (PlacementID, InterviewID, Position)

## Step-5: Normalizing the database Schema

Normalization is a process to reduce redundancy in the database and minimize the insert, delete and update operations without having any abnormalities.

### FIRST NORMAL FORM

Every table in the above database is having the

1. Primary key.
2. All the values in each column of the table are atomic.

### SECOND NORMAL FORM

All the tables in the database are in 1NF. All the non-key attributes of the table are fully dependent on the primary key of the table and there should not be any partial redundancy in the table. Here in the database, no tables contain partial redundancy and all the non-key attributes of the table are completely dependent on the primary key of the table.

### THIRD NORMAL FORM

All the tables in the database are:

-> 1NF

-> 2NF

-> No transitivity (A -> B -> C) in any of the tables of the database.

Note: All the tables in our database are in normalized form.

## Step-6: Create a database with constraints

Creating Students table

```
1 CREATE TABLE Students (  
2     StudentID INT PRIMARY KEY,  
3     FName VARCHAR(50),  
4     LName VARCHAR(50),  
5     Gender ENUM('Male', 'Female'),  
6     Department VARCHAR(30),  
7     DateOfBirth DATE,  
8     Email VARCHAR(100),  
9     Phone BIGINT  
10 );
```

---

### Creating Companies table

```
1 CREATE TABLE Companies (  
2   CompanyID INT PRIMARY KEY,  
3   CompanyName VARCHAR(100),  
4   Domain VARCHAR(50),  
5   Branch VARCHAR(100),  
6   Website VARCHAR(100)  
7 );
```

### Creating Skills table

```
1 CREATE TABLE Skills (  
2   SkillID INT PRIMARY KEY,  
3   SkillName VARCHAR(50)  
4 );
```

### Creating StudentSkills table

```
1 CREATE TABLE StudentSkills (  
2   StudentID INT,  
3   SkillID INT,  
4   PRIMARY KEY (StudentID, SkillID),  
5   FOREIGN KEY (StudentID) REFERENCES Students(StudentID),  
6   FOREIGN KEY (SkillID) REFERENCES Skills(SkillID)  
7 );
```

### Creating CompanyReqSkill table

```
1 CREATE TABLE CompanyReqSkills (  
2   CompanyID INT,  
3   SkillID INT,  
4   PRIMARY KEY (CompanyID, SkillID),  
5   FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID),  
6   FOREIGN KEY (SkillID) REFERENCES Skills(SkillID)  
7 );
```

### Creating Interviews table

```
1 CREATE TABLE Interviews (  
2   InterviewID INT PRIMARY KEY,  
3   StudentID INT,  
4   CompanyID INT,  
5   InterviewDate DATE,  
6   InterviewResult ENUM( Selected , Rejected , To be determine  
7   FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID),  
8   FOREIGN KEY (StudentID) REFERENCES Students(StudentID)  
9 );
```

### Creating Placements table

```
1 CREATE TABLE Placements (  
2   PlacementID INT PRIMARY KEY AUTO_INCREMENT ,  
3   InterviewID INT,  
4   FOREIGN KEY (InterviewID) REFERENCES Interviews(InterviewID)  
5   Position VARCHAR(100)  
6 );
```

## Step-7: Inserting values into the tables

### Inserting Values Into Students table

```
1 INSERT INTO Students Values (01,'Rahul','S','Male','MCA','1995-10-10','Rahul.  
   S@gmail.com',1234567890);
```

---

```

2 INSERT INTO Students Values (02, 'Muskan', 'T', 'Female', 'MCA', '2002-02-15', 'Muskan.
  T@gmail.com', 9133566702);
3 INSERT INTO Students Values (03, 'Guru', 'D', 'Male', 'MCA', '2002-12-15', 'Guru.
  D@gmail.com', 9123896789);
4 INSERT INTO Students Values (04, 'Pratik', 'K', 'Male', 'MCA', '2002-01-05', 'Pratik.
  K@gmail.com', 9176258463);
5 INSERT INTO Students Values (05, 'Sayli', 'M', 'Female', 'MCA', '2001-05-04', 'Sayli.
  M@gmail.com', 9147652384);
6 INSERT INTO Students Values (07, 'Pratik', 'R', 'Male', 'MCA', '2001-05-04', 'Pratik.
  R@gmail.com', 9147782384);
7 INSERT INTO Students Values (08, 'Atharva', 'K', 'Male', 'MCA', '2001-01-07', 'Atharva.
  K@gmail.com', 9147852384);
8 INSERT INTO Students Values (12, 'Anuja', 'C', 'Female', 'MCA', '1999-09-28', 'Anuja.
  S@gmail.com', 9147652384);
9 INSERT INTO Students Values (41, 'Darshan', 'P', 'Male', 'MCA', '2002-12-12', '
  pathakdarshan12@gmail.com', 9389745620);
10 INSERT INTO Students Values (47, 'Prabhujyoti', 'S', 'Female', 'MCA', '2001-02-19', '
  prabhujyoti.s@gmail.com', 9847526581);

```

### Inserting Values Into Companies table

```

1 INSERT INTO Companies VALUES (01, 'FaceBook', 'Information Technology', 'New York
  ', 'www.facebook.com');
2 INSERT INTO Companies VALUES (02, 'Amazon', 'Information Technology', 'New York',
  'www.Amazon.in');
3 INSERT INTO Companies VALUES (03, 'Apple', 'Information Technology', 'New York',
  'www.apple.com');
4 INSERT INTO Companies VALUES (04, 'Netflix', 'Information Technology', 'New York'
  , 'www.netflix.com');
5 INSERT INTO Companies VALUES (05, 'Google', 'Information Technology', 'New York',
  'www.google.com');
6 INSERT INTO Companies VALUES (06, 'IBM', 'Information Technology', 'New York', '
  www.ibm.com');
7 INSERT INTO Companies VALUES (07, 'Microsoft', 'Information Technology', 'Redmond
  ', 'www.microsoft.com');
8 INSERT INTO Companies VALUES (08, 'Tesla', 'Automotive', 'Palo Alto', 'www.tesla.
  com');
9 INSERT INTO Companies VALUES (09, 'Alphabet Inc.', 'Information Technology', '
  Mountain View', 'www.abc.xyz');
10 INSERT INTO Companies VALUES (10, 'Walmart', 'Retail', 'Bentonville', 'www.
  walmart.com');

```

### Inserting Values Into Skills table

```

1 INSERT INTO Skills VALUES (1, 'JAVA');
2 INSERT INTO Skills VALUES (2, 'Python');
3 INSERT INTO Skills VALUES (3, 'C++');
4 INSERT INTO Skills VALUES (4, 'Full Stack');
5 INSERT INTO Skills VALUES (5, 'SQL');
6 INSERT INTO Skills VALUES (6, 'Leadership');
7 INSERT INTO Skills VALUES (7, 'Team Management');
8 INSERT INTO Skills VALUES (8, 'Machine Learning');
9 INSERT INTO Skills VALUES (9, 'Data Analysis');
10 INSERT INTO Skills VALUES (10, 'Project Management');

```

### Inserting Values Into StudentSkills table

```

1 INSERT INTO StudentSkills VALUES (01,1);
2 INSERT INTO StudentSkills VALUES (02,2);
3 INSERT INTO StudentSkills VALUES (03,1);
4 INSERT INTO StudentSkills VALUES (03,3);
5 INSERT INTO StudentSkills VALUES (04,3);
6 INSERT INTO StudentSkills VALUES (04,4);
7 INSERT INTO StudentSkills VALUES (05,2);
8 INSERT INTO StudentSkills VALUES (05,5);
9 INSERT INTO StudentSkills VALUES (41,5);

```

---

```

10 INSERT INTO StudentSkills VALUES (41,8);
11 INSERT INTO StudentSkills VALUES (41,9);
12 INSERT INTO StudentSkills VALUES (47,2);
13 INSERT INTO StudentSkills VALUES (47,6);
14 INSERT INTO StudentSkills VALUES (47,7);

```

### Inserting Values Into CompanyReqSkills table

```

1 INSERT INTO CompanyReqSkills VALUES (01,1);
2 INSERT INTO CompanyReqSkills VALUES (02,2);
3 INSERT INTO CompanyReqSkills VALUES (03,1);
4 INSERT INTO CompanyReqSkills VALUES (03,3);
5 INSERT INTO CompanyReqSkills VALUES (04,2);
6 INSERT INTO CompanyReqSkills VALUES (05,3);
7 INSERT INTO CompanyReqSkills VALUES (05,5);
8 INSERT INTO CompanyReqSkills VALUES (06,5);
9 INSERT INTO CompanyReqSkills VALUES (06,8);
10 INSERT INTO CompanyReqSkills VALUES (06,9);

```

Inserting Values Into Interviews table Note: Inserting values into interviews will automatically call the trigger MatchandPlace which will place the students in Placements table if the skill acquired by students and skills required by the company matches.

```

1 CREATE TABLE Interviews (
2   InterviewID INT PRIMARY KEY,
3   StudentID INT,
4   CompanyID INT,
5   InterviewDate DATE,
6   InterviewResult ENUM( Selected , Rejected , To be determine
7   FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID),
8   FOREIGN KEY (StudentID) REFERENCES Students(StudentID)
9 );

```

### Creating Trigger MatchAndPlace

Note: Inserting values into interviews will automatically call the trigger MatchandPlace this trigger is used to place students into Placement table if the skills acquired by students match all the skill requirements of the company for a particular interview.

```

1 CREATE TRIGGER MatchAndPlace
2 AFTER INSERT ON Interviews
3 FOR EACH ROW
4 BEGIN
5   DECLARE student_skill_count INT;
6   DECLARE matching_skill_count INT;
7   DECLARE ID DATE;
8
9   -- Count the total number of skills for the student
10  SELECT COUNT(*) INTO student_skill_count
11  FROM StudentSkills
12  WHERE StudentID = NEW.StudentID;
13
14  -- Count the number of matching skills between the student and the company
15  SELECT COUNT(*) INTO matching_skill_count
16  FROM StudentSkills AS SS
17  JOIN CompanyReqSkills AS CS ON SS.SkillID = CS.SkillID
18  WHERE SS.StudentID = NEW.StudentID AND CS.CompanyID = NEW.CompanyID;
19
20  -- If all the student's skills match with the company's requirements
21  IF student_skill_count = matching_skill_count THEN
22    INSERT INTO Placements (InterviewID, Position)
23    VALUES (New.InterviewID, 'To be determined');
24  END IF;
25 END;
26 //
27 DELIMITER ;

```



## Step-8: Database tables

### Skills

	SkillID	SkillName
▶	1	JAVA
	2	Python
	3	C++
	4	Full Stack
	5	SQL
	6	Leadership
	7	Team Management
	8	Machine Learning
	9	Data Analysis
	10	Project Management
•	NULL	NULL

### Students

	StudentID	FName	LName	Gender	Department	DateOfBirth	Email
▶	1	Rahul	S	Male	MCA	1995-10-10	Rahul.S@gmail.com
	2	Muskan	T	Female	MCA	2002-02-15	Muskan.T@gmail.com
	3	Guru	D	Male	MCA	2002-12-15	Guru.D@gmail.com
	4	Pratik	K	Male	MCA	2002-01-05	Pratik.K@gmail.com
	5	Sayli	M	Female	MCA	2001-05-04	Sayli.M@gmail.com
	7	Pratik	R	Male	MCA	2001-05-04	Pratik.R@gmail.com
	8	Atharva	K	Male	MCA	2001-01-07	Atharva.K@gmail.com
	12	Anuja	C	Female	MCA	1999-09-28	Anuja.S@gmail.com
	37	Darshani	M	Female	MCA	2002-08-01	darshani.M@gmail.com
	41	Darshan	P	Male	MCA	2002-12-12	pathakdarshan12@gmail.com
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## StudentSkills

	StudentID	SkillID
▶	1	1
	2	1
	3	1
	4	1
	5	1
	2	2
	3	2
	4	2
	5	2
	3	3
★	NULL	NULL

## Companies

	CompanyID	CompanyName	Domain	Branch	Website
▶	1	FaceBook	Information Technology	New York	www.facebook.com
	2	Amazon	Information Technology	New York	www.Amazon.in
	3	Apple	Information Technology	New York	www.apple.com
	4	Netflix	Information Technology	New York	www.netflix.com
	5	Google	Information Technology	New York	www.google.com
	6	IBM	Information Technology	New York	www.ibm.com
	7	Microsoft	Information Technology	Redmond	www.microsoft.com
	8	Tesla	Automotive	Palo Alto	www.tesla.com
	9	Alphabet Inc.	Information Technology	Mountain View	www.abc.xyz
	10	Alibaba Group	E-commerce	Hangzhou	www.alibaba.com
★	NULL	NULL	NULL	NULL	NULL

---

## CompanyReqSkills

	CompanyID	SkillID
▶	1	1
	2	1
	3	1
	5	1
	2	2
	4	2
	3	3
	5	3
	4	4
	5	5
●	NULL	NULL

## Interviews

	InterviewID	StudentID	CompanyID	InterviewDate	InterviewResult
▶	1	3	3	2023-08-15	Selected
	2	2	4	2023-10-28	Selected
	3	3	5	2023-10-25	Selected
	4	3	1	2023-04-08	Selected
	5	3	3	2023-04-15	Selected
	6	3	5	2023-12-15	Selected
	7	41	6	2023-06-11	Selected
●	NULL	NULL	NULL	NULL	NULL

---

## Placements

	PlacementID	InterviewID	Position
▶	1	7	To be determined
	2	101	To be determined
	3	102	To be determined
	4	103	To be determined
	5	104	To be determined
	6	105	To be determined
	7	106	To be determined
	8	107	To be determined
✱	NULL	NULL	NULL

### Step-9: Queries based on Model

```
1 -- 1. Selecting Students Where Name Starts With D
2 SELECT * FROM Students WHERE FNAME LIKE 'D%';
3
4 -- 2. Selecting Students whose date of birth is between 1985-01-01 and
   2000-12-31
5 SELECT * FROM Students WHERE DateOfBirth BETWEEN '1985-01-01' AND '2000-12-31';
6
7 -- 3. Selecting students of MCA department
8 SELECT * FROM Students WHERE Department LIKE 'MCA';
9
10 -- 4. Selecting female students from table
11 SELECT * FROM Students WHERE Gender = 'Female';
12
13 -- 5. Selecting company names starting with A or F
14 SELECT * FROM Companies WHERE CompanyName LIKE 'A%' OR CompanyName LIKE 'F%';
15
16 -- 6.
17 -- Counting Male Students
18 SELECT COUNT(StudentID) FROM Students Where Gender LIKE 'Male';
19 -- Counting Female Students
20 SELECT COUNT(StudentID) FROM Students Where Gender LIKE 'Female';
21
22 -- 7. Selecting CompanyNames in Uppercase
23 SELECT UPPER(CompanyName) FROM Companies;
24
25 -- 8. Concatinating first name and last name of students
26 SELECT CONCAT(FName, " ", LName) AS FullName FROM Students;
27
28 -- 9. Selecting Interviews where interviewdate is 2000-01-01 and 2023-12-30
29 SELECT * FROM Interviews WHERE InterViewDate BETWEEN '2000-01-01' AND '2023-12-30
   ' ;
30
31 -- 10.
32 -- Selecting companies of domain Information tecnology
```

---

```

33 SELECT * FROM Companies WHERE Domain='Information Technology';
34
35 -- Selecting companies of branch new york
36 SELECT * FROM Companies WHERE Branch='New York';
37
38 -- 11. students who have no placements
39 SELECT StudentID ,FName
40 FROM Students
41 WHERE StudentID not in
42 (SELECT DISTINCT (StudentID) FROM Placements);
43
44 -- 12.students who had given interview at least one time
45 SELECT DISTINCT(StudentID)
46 FROM Interviews
47 WHERE StudentID IN
48 (SELECT DISTINCT (StudentID) FROM Interviews);
49
50 -- 13.students who are placed in company
51 SELECT I.StudentID
52 FROM Placements P
53 JOIN Interviews I ON I.InterviewID = P.InterviewID
54 WHERE CompanyID='02';
55
56 -- 14.Counting Number of Interviews given by Student
57 SELECT S.StudentID, S.fname, COUNT(I.InterviewID) AS InterviewCount
58 FROM Students S
59 LEFT JOIN Interviews I ON S.StudentID = I.StudentID
60 GROUP BY S.StudentID, S.fname;
61
62
63 -- 15. how many skills student have
64 SELECT S.StudentID,S.FName, COUNT(SS.SkillID) AS SkillsCount
65 FROM Students S , StudentSkills SS
66 WHERE S.StudentID = SS.StudentID GROUP BY StudentID;
67
68 SELECT DISTINCT(C.CompanyID),C.CompanyName,COUNT(CS.SkillID) AS SkillsReqCount
69 FROM Companies C , CompanyReqSkills CS
70 WHERE C.CompanyID = CS.CompanyID GROUP BY CompanyID;
71
72 -- 16. students having more than 2 skills
73 SELECT S.StudentID, S.FName,COUNT(SS.SkillID) AS SkillsCount
74 FROM Students S , StudentSkills SS
75 WHERE S.StudentID = SS.StudentID GROUP BY StudentID
76 HAVING COUNT(SS.SkillID) > 2;
77
78 -- 17.company requires less than 3 skills
79 SELECT C.CompanyID, C.CompanyName,COUNT(CS.SkillID) AS ReqSkillCount
80 FROM Companies C , CompanyReqSkills CS
81 WHERE C.CompanyID = CS.CompanyID GROUP BY CompanyID
82 HAVING COUNT(CS.SkillID) < 3;
83
84 -- 18. students having interview result as selected
85 select S.StudentID, S.FName , C.CompanyID, C.CompanyName ,I.InterviewResult
86 from Interviews I , Students S , Companies C
87 Where I.StudentID = S.StudentID AND I.CompanyID = C.CompanyID GROUP BY I.
InterviewID
88 HAVING I.InterviewResult = 'Selected';
89
90 -- 19.
91 -- Counting Number of Students
92 SELECT COUNT(DISTINCT(StudentID))
93 FROM Students;
94

```

---

---

```

95 -- Counting Number of Companies
96 SELECT COUNT(DISTINCT(CompanyID))
97 FROM Companies;
98
99 -- 20. students who born before 1st jan 2000
100 SELECT S.StudentID, S.FName, S.DateOfBirth
101 FROM Students S
102 WHERE DateOfBirth < '2000-01-01';
103
104 -- 21.
105 -- Creating Query To View All The Placement Related Data
106 DELIMITER //
107 CREATE PROCEDURE GetPlacementData()
108 BEGIN
109     SELECT
110         S.StudentID,
111         S.FName,
112         S.LName,
113         S.Gender,
114         S.Department,
115         S.DateOfBirth,
116         S.Email,
117         S.Phone,
118         C.CompanyID,
119         C.CompanyName,
120         C.Domain,
121         C.Branch,
122         C.Website,
123         I.InterviewID,
124         I.InterviewDate,
125         I.InterviewResult,
126         P.Position
127 FROM Placements AS P
128 JOIN Interviews AS I ON P.InterviewID = I.InterviewID
129 JOIN Students AS S ON S.StudentID = I.StudentID
130 JOIN Companies AS C ON C.CompanyID = I.CompanyID
131 WHERE P.InterviewID = I.InterviewID;
132 END;
133 //
134 DELIMITER ;
135 CALL GetPlacementData();
136
137 -- 22. Query to Get Data of students matching skills with company
138 SELECT SS.StudentID, S.fname, C.CompanyID, C.CompanyName, Sks.SkillID, Sks.SkillName
139 FROM StudentSkills SS
140 JOIN Students S ON SS.StudentID = S.StudentID
141 JOIN CompanyReqSkills CRS ON SS.SkillID = CRS.SkillID
142 JOIN Companies C ON CRS.CompanyID = C.CompanyID
143 JOIN Skills SKs ON SS.SkillID=CRS.SkillID=SKs.SkillID
144 GROUP BY SS.StudentID, S.fname, C.CompanyID, C.CompanyName, Sks.SkillID, Sks.
    SkillName
145 ORDER BY S.StudentID;
146
147 -- 23. List the students who have been selected in interviews and the positions
    they were offered:
148 SELECT S.FName, S.LName, P.Position
149 FROM Students S
150 JOIN Interviews I ON S.StudentID = I.StudentID
151 JOIN Placements P ON I.InterviewID = P.InterviewID
152 WHERE I.InterviewResult = 'Selected';
153
154
155 -- 24. Count the number of interviews for each company:

```

---

---

```
156  SELECT C.CompanyName, COUNT(I.InterviewID) AS InterviewCount
157  FROM Companies C
158  LEFT JOIN Interviews I ON C.CompanyID = I.CompanyID
159  GROUP BY C.CompanyName;
160
161  -- 25.List the students who have not acquired any skills:
162  SELECT S.FName, S.LName
163  FROM Students S
164  LEFT JOIN StudentSkills SS ON S.StudentID = SS.StudentID
165  WHERE SS.SkillID IS NULL;
```

## Conclusion

The development of the Placement Management System, as described in this case study, is an important step towards enhancing our college's efficiency in helping students secure job placements. The database design and normalization ensure data accuracy and consistency, while the implementation of MySQL, along with the use of triggers, offers a practical and efficient way to manage the placement process. This system serves as a valuable tool in the modern educational landscape, facilitating student success and college achievements.